

08-01-00

A

IBM Docket No. GB9-2000-0083-US

# In the United States Patent and Trademark Office

## Patent Application Transmittal

Transmitted herewith for filing is the Patent Application of:

Inventors(s): Michael Casson Bailey, Timothy William Banks, Geoffrey Redding, and Dennis J. Zimmer  
 For: Method, Program Product and Computer System for Progressive Improvement of an Environment Pool

### Enclosed are

20 pages of specification, including 18 claims, plus 2 sheets of drawings.  
 An assignment of the invention to International Business Machines Corporation, Armonk, New York 10504.

A certified copy of a/an application.

X Declaration and Power of Attorney. Copy, Not Executed

PTO-1449 & references

x A return post card

Other:

### Filing Fee Calculation (For Other Than Small Entity)

Basic Fee:							\$690.00
Claims Fees:		Filed	Limit	Extra		Rate per Extra	
Total claims:		18	20	0		\$18.00	\$0.00
Independent claims:		2	3	0		\$78.00	\$0.00
	Multiple Dependent Claim Presented					\$260.00	\$0.00
Total							\$690.00

Please charge Deposit Account 09-0461 for the Total set forth above. The Commissioner is authorized to charge payment of any additional filing fees required under 37 CFR §1.16 and any patent application processing fees under 37 CFR §1.17 or to credit any overpayment to the identified account. A duplicate copy of this sheet is enclosed.

### Express Mail Certificate

Express Mail Label No: EK873465157US

Date: July 31, 2000

I hereby certify that I am depositing the papers identified above with the U.S. Postal Service "Express Mail Post Office to Address" service on the above date, addressed to the Commissioner of Patents and Trademarks, Washington, DC 20231

*Linda Dupont*  
 Linda Dupont

BY:

*Edward H. Duffield*  
 Edward H. Duffield

Attorney of Record Reg. No. 25,970

Date: July 31, 2000

IBM Corporation T81/062  
 Intellectual Property Law  
 PO Box 12195  
 Res. Tri. Park, NC 27709

Telephone: 919-254-1301 FAX 919-254-4330

**METHOD, PROGRAM PRODUCT AND COMPUTER SYSTEM FOR PROGRESSIVE  
IMPROVEMENT OF AN ENVIRONMENT POOL**

FIELD OF THE INVENTION

The present invention relates to the provision of preinitialized environments to support multiple programs executing in a computer system with a reduction in program task initialization time.

BACKGROUND OF THE INVENTION

In computer systems, it is well known in the art to have multiple programs and instances of the same program running at the same time. Indeed, many of the problems and difficulties of managing modern systems have come about precisely because the information technology world has advanced since the days of dedicated batch processing systems in which only a single computing task could be executed at any given time.

One problem that is particularly acute in modern systems is the impact on performance of initializing runtime environments.

Various approaches have been taken to provide a solution to this problem. One example is that described in

US Patent No. 5,896,531 to Curtis et al., in which it is proposed to retain environments that have been initialized to form a pool of environments for future use. In the model proposed by Curtis et al., a second application program's initiation causes the system to examine existing environments for possible reuse if one of them should prove to meet the minimal requirements of the newly-initiated program. Should an examined environment prove not to meet the minimal requirements, it is discarded from the set of existing environments.

The minimal requirements disclosed in Curtis et al. are a set of the parameters required by object-oriented programs for their execution in an object-oriented program system.

In a system as proposed by Curtis et al., a trade-off is found between the saving of initialization time by keeping environments for reuse and the consumption of storage by the system because of the number of preinitialized environments that are in memory at any time. The trade-off is achieved by the expedient of discarding preinitialized environments that are found not to fit the second application's minimal requirements.

In such a system, there is an element of wastage, in that environments are discarded which might in time have

proven to be useful for subsequent application programs to reuse. Further, the use of environments from the environment pool is limited to those environments which meet the minimal requirements of the second or subsequent application.

It is desirable to mitigate these problems by finding some more economical way to control the reuse of environments in a system in which multiple programs and instances of programs may coexist.

#### SUMMARY OF THE INVENTION

Accordingly, in a first aspect, the present invention provides a method for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, the method comprising steps of: providing a first environment for a first program; responsive to initiation of a second program, making a determination whether creation of a new environment is a best response; responsive to a determination that creation of a new environment is a best response, creating a new environment for the second program; responsive to a determination that creating a new environment is not a best response, testing the pool for a best fit environment; and adding elements to the best fit environment to match requirements of the second program, unless the best fit

environment already matches the requirements of the second program.

Preferably, at least one of the first, new and best fit environments is an execution environment. Preferably,  
5 the execution environment is preinitialized.

Preferably at least one of the first, new and best fit environments is eligible to be deleted. Further preferably, at least one least recently used of the first, new and best  
10 fit environments is eligible to be deleted.

Preferably, the elements are parameters of at least one of the first, the new and the best fit environments.

The step of responsive to initiation of a second program, making a determination whether creation of a new environment is a best response preferably comprises testing whether the pool has reached a maximum size. Further preferably, the step of responsive to a determination that the pool has reached its maximum size, testing the pool for a best fit environment comprises a programmatically alterable test.

In a second aspect, the present invention provides a  
25 computer program product comprising computer program code tangibly embodied in a signal-bearing medium, for, when loaded into a computer system and executed, progressively

improving a fit of a pool of reusable environments to requirements of programs in a computer system, by causing the computer system to perform the steps of a method according to the first aspect. The signal bearing medium is preferably at least one of a transmissive medium and a storage medium.

In a third aspect, the present invention provides a computer system for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, the computer system comprising: means for providing a first environment for a first program; means responsive to initiation of a second program, for making a determination whether creation of a new environment is a best response; means responsive to a determination that creation of a new environment is a best response, for creating a new environment for the second program; means responsive to a determination that creating a new environment is not a best response, for testing the pool for a best fit environment; and means for adding elements to the best fit environment to match requirements of the second program, unless the best fit environment already matches the requirements of the second program.

The preferred features of the means of the third aspect correspond to the preferred features of the steps of a method according to the first aspect.

It is known in the programming art to use preinitialized environments when an environment, such as an execution environment for a program, is expensive (in system time and resource consumption) to build, and there is a performance benefit to be gained by preserving the environment and reusing it for subsequent requests. In preferred embodiments of the present invention, several instances of a preinitialized environment can be made, different from each other depending on which program was loaded and executed in each.

In a preferred embodiment of the present invention, parameters are passed to the allocator of environments, which parameters are used to select the most appropriate environment. If a new environment is created, the matching parameters are saved with it. When selecting an environment to allocate, the allocator chooses an environment with matching parameters. If the allocator is obliged to choose an environment which does not match, it adds the matching parameters to the set for that environment, as the environment will then acquire the resources that those parameters describe, and so it will be a better match in future, providing that future programs require that set of parameters or a subset thereof.

When an environment has aged, the system may preferably destroy it, so allowing the requests profile to change over time, and the preinitialized environments to follow.

5

Preinitialized environments grow as they acquire resources. There may be advantages to keeping them as small as possible, and this can be achieved by using them for the same purpose as much as possible, and reducing the likelihood of all environments being used for all purposes. The method of the preferred embodiment of the present invention is advantageously flexible in allowing the system to learn the kinds of environments needed and reuse them efficiently.

10

A further advantage of the preferred embodiment of the present invention and its preferred features is that the pool of preinitialized environments is capable of more purposeful and less wasteful change than the prior art environments, in that environments that are not optimal are given further chances to become so. Furthermore, as a workload changes over time, the preferred embodiment of the present invention and its preferred features permit the corresponding adaptation and readaptation of the pool.

20

25

An additional advantage of the preferred embodiment of the present invention and its preferred features is in the



programmably adaptable nature of the matching algorithms that can be used. This allows a system programmer, for example, to adjust the criteria on which a best-fit judgement will be made.

5

#### BRIEF DESCRIPTION OF THE DRAWINGS

A preferred embodiment of the invention will now be described by way of example only, with reference to the accompanying drawings, in which:

10

Figure 1 shows a computer system suitable to be used as a preferred embodiment of the present invention.

Figure 2 shows the steps of a method according to a preferred embodiment of the present invention.

#### DETAILED DESCRIPTION OF THE INVENTION

In Figure 1 there is shown a computer system (100), having a processor (102), a memory (104) and auxiliary storage (106) which may comprise magnetic storage, optical storage or the like. As a program (108) is initiated, one of environments (116, 118) it requires is instantiated in memory (104). As is well known to those skilled in the art, programs and data may be paged into and out of memory (104) into storage (106) during processing, but in this

25

description, these environments will be referenced as if they remained in memory (104).

5        Within memory (104) are shown a program (108) and a  
pool (114) which may contain preinitialized environments  
(116, 118). In the case of a first initiation of a program  
(108), typically there are no environments available, and  
the first program must bear the cost of initialization of  
an environment. In the case of a second or subsequent  
10    program (122), there may be preinitialized environments.  
In a preferred embodiment, a dispatcher (124) is the  
subsystem of the system which allocates environments for  
use by programs. Dispatcher (124) may be programmed, for  
example, to offer preinitialized environments for reuse by  
any second or subsequent program (122). In a preferred  
embodiment, earlier programs are forced, in circumstances  
to be described in the description of the method below, by  
dispatcher (124) to accept an altruistic role so long as  
there is room in the pool (114) for a new environment  
20    (126). In this way, a subset of the earlier programs must  
bear the cost in performance and system resource use of  
initializing environments, for the benefit of later  
programs or instances of programs. This altruistic  
behavior allows the system to build up a pool (114) of  
25    environments which is conveniently sized to optimize the  
consumption of resources within the system, while offering  
programs an increased chance that there will be a suitable

environment, or one that can be made suitable by addition to the set of elements (128, 130, 132)) contained in an environment. In one embodiment, elements (128, 130, 132) are parameters representing resources available to the program, but those skilled in the art will understand that they could equally well be other environment elements such as program control settings.

The preferred embodiment of the present invention is as a subsystem within a CICS Transaction Server (CICS is a registered trademark of IBM Corporation). In the preferred embodiment, the environments that form the pool are specialized open Task Control Blocks (open TCBs) that are able to provide the enclave (or task runtime language environment) for instances of applications within the system. These open TCBs are reserved for use by Java (Java is a registered trademark of Sun Microsystems, Inc.) program objects within the system, and are limited in number in order to keep system resource usage within control. The CICS Dispatcher acts as the allocator of environments to act as runtime language environments for use by the Java program objects.

In Figure 2 are shown the steps of a method according to a preferred embodiment of the present invention.

In step (200) a program is initiated and determined to be a Java program object suitable to use a preinitialized environment from a pool. The starting parameter elements may be, for example, as shown in the exemplary drawing, **program** and **transid**. The dispatcher tests at step (204) if there is a free environment which matches both the parameter elements **program** and **transid**. If there is, that environment will be used at step (224). If there is not, the dispatcher tests at step (206), whether the best response to the situation is to create a new environment. In one embodiment, the test may be based on a maximum size setting for the environment pool. If the maximum size has been reached, it is a best response to attempt to locate a nearly-matching preinitialized environment. Other possible criteria will be evident to those skilled in the art -- for example, it may be better in some cases to stall certain program tasks until a free environment becomes available.

If the dispatcher determines that the best response is to create a new environment, it initiates that process at step (208) and causes the required parameter elements to be added to the environment. That environment is then used at step (224). If the dispatcher determines that the creation of a new environment is not the best response, it tests at step (210) for a partially matching environment in the pool. If there is one, it causes the required parameter

elements to be added to the environment. That environment is then used at step (224).

5 If the dispatcher determines that no partially  
matching environment is available in the pool, it tests at  
step (214) for a free environment in the pool that does not  
match at all. If there is one, it causes the required  
parameter elements which are currently missing from the  
environment to be added to the environment at step (216).  
10 That environment is then used at step (224).

If the dispatcher determines that no free environment  
is available in the pool, it tests at step (218) for a free  
environment in another pool. If there is one, the  
dispatcher at step (220) steals it from that pool and  
causes the required parameter elements to be added to the  
environment. That environment is then used at step (224).

20 In the event of a complete failure to create a new  
environment or locate a preinitialized environment, the  
dispatcher at step (222) suspends the program task, and at  
the end of the suspension period, returns to the beginning  
of the process to begin to retry the process from step  
(204).

25 It will thus be clear to one skilled in the art that,  
over a period of time running a workload, the pooled

environments will "learn" the characteristics of the workload's program tasks. At some point, the number of environments may need to be trimmed to remove environments that can be taken to be no longer needed. In one  
5 embodiment of the present invention, it is possible to use a least-recently-used algorithm to decide to remove what may be environments that are no longer needed. Those skilled in the art will recognise that other methods of deciding which environments are no longer needed may be  
10 used. One example would be the use of frequency counts, which are well known in the art. If a single program task is instantiated very frequently, its preinitialized environments will most likely remain available in the pool. In an embodiment having a least-recently-used algorithm controlling the trimming of environments, if a task is infrequently run, its preinitialized environments will most likely be purged so that they do not remain available in the pool. As the characteristics of the program tasks in a workload change over time, so will the content of the pool tend to modify itself to match, by a kind of Darwinian evolution, in which the environments that are best modified to satisfy the requirements of the program tasks in the workload will survive.

25 The present invention is preferably embodied as a computer program product for use with a computer system. Such an implementation may comprise a series of computer

readable instructions either fixed on a tangible medium, such as a computer readable medium, e.g., diskette, CD-ROM, ROM, or hard disk, or transmittable to a computer system, via a modem or other interface device, over either a  
5 tangible medium, including but not limited to optical or analog communications lines, or intangibly using wireless techniques, including but not limited to microwave, infrared or other transmission techniques. The series of computer readable instructions embodies all or part of the  
10 functionality previously described herein.

Those skilled in the art will appreciate that such computer readable instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Further, such instructions may be stored using any memory technology, present or future, including but not limited to, semiconductor, magnetic, or optical, or transmitted using any communications technology, present or future, including but not limited to optical, infrared, or microwave. It is contemplated that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation, e.g., shrink wrapped software, pre-loaded with a computer system, e.g., on a system ROM or fixed disk, or distributed from a server or electronic  
20 bulletin board over a network, e.g., the Internet or World Wide Web.

**CLAIMS**

What is claimed is:

5

1. A method for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, the method comprising steps of:

10 providing a first environment for a first program;

responsive to initiation of a second program, making a determination whether creation of a new environment is a best response;

responsive to a determination that creation of a new environment is a best response, creating a new environment for the second program;

responsive to a determination that creating a new environment is not a best response, testing the pool for a best fit environment; and

25

adding elements to the best fit environment to match requirements of the second program, unless the best fit environment already matches the requirements of the second program.



2. The method of claim 1, wherein at least one of the first, new and best fit environments is an execution environment.

5 3. The method of claim 2, wherein the execution environment is preinitialized.

10 4. The method of claim 1, wherein at least one of the first, new and best fit environments is eligible to be deleted.

15 5. The method of claim 4, wherein at least one least recently used of the first, new and best fit environments is eligible to be deleted.

20 6. The method of claim 1, wherein the elements are parameters of at least one of the first, the new and the best fit environments.

25 7. The method of claim 1, wherein the step of responsive to initiation of a second program, making a determination whether creation of a new environment is a best response comprises testing whether the pool has reached a maximum size.

8. The method of claim 7, wherein the step of responsive to a determination that the pool has reached its maximum

size, testing the pool for a best fit environment comprises a programmatically alterable test.

5 9. A computer program product, comprising computer program code tangibly embodied in a signal-bearing medium, for, when loaded into a computer system and executed, progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, by causing the computer system to perform the steps  
10 of a method as claimed in claim 1.

10. A computer program product as claimed in claim 9, wherein the signal bearing medium is at least one of a transmissive medium and a storage medium.

11. A computer system for progressively improving a fit of a pool of reusable environments to requirements of programs in a computer system, the computer system comprising:

means for providing a first environment for a first program;

means responsive to initiation of a second program, for making a determination whether creation of a new environment is a best response;

25

means responsive to a determination that creation of a new environment is a best response, for creating a new environment for the second program;

5 means responsive to a determination that creating a new environment is not a best response, for testing the pool for a best fit environment; and

10 means for adding elements to the best fit environment to match requirements of the second program, unless the best fit environment already matches the requirements of the second program.

12. The computer system of claim 11, wherein at least one of the first, new and best fit environments is an execution environment.

13. The computer system of claim 12, wherein the execution environment is preinitialized.

14. The computer system of claim 11, wherein at least one of the first, new and best fit environments is eligible to be deleted.

25 15. The computer system of claim 14, wherein at least one least recently used of the first, new and best fit environments is eligible to be deleted.

5

10

Figure 1 displays 15 bar charts showing the percentage of respondents for various demographic and attitudinal variables. The variables are: 1. Age, 2. Sex, 3. Education, 4. Income, 5. Employment, 6. Religion, 7. Political Party, 8. Marital Status, 9. Number of Children, 10. Number of Siblings, 11. Number of Pets, 12. Number of Vehicles, 13. Number of Telephones, 14. Number of Computers, 15. Number of Televisions. Each chart has a y-axis from 0 to 100% and an x-axis with categories for each variable. The data is as follows:

Variable	Category	Percentage
1. Age	18-24	15%
	25-34	25%
	35-44	30%
	45-54	20%
	55-64	10%
2. Sex	Male	55%
	Female	45%
3. Education	High School	35%
	College	45%
	Graduate	20%
4. Income	\$0-\$10,000	15%
	\$10,000-\$20,000	30%
	\$20,000-\$30,000	25%
	\$30,000-\$40,000	20%
	\$40,000-\$50,000	10%
5. Employment	Full-time	45%
	Part-time	30%
	Unemployed	25%
	Retired	10%
6. Religion	Catholic	45%
	Protestant	35%
	Jewish	10%
	Muslim	5%
	Other	5%
7. Political Party	Democrat	55%
	Republican	40%
8. Marital Status	Married	65%
	Single	25%
9. Number of Children	0	35%
	1	25%
10. Number of Siblings	0	15%
	1	25%
	2	30%
	3	20%
	4	10%
11. Number of Pets	0	15%
	1	30%
	2	25%
	3	20%
	4	10%
12. Number of Vehicles	0	15%
	1	30%
	2	25%
	3	20%
	4	10%
13. Number of Telephones	0	15%
	1	30%
	2	25%
	3	20%
	4	10%
14. Number of Computers	0	15%
	1	30%
	2	25%
	3	20%
	4	10%
15. Number of Televisions	0	15%
	1	30%
	2	25%
	3	20%
	4	10%

**ABSTRACT**

Disclosed are a method, system and computer program product  
for progressively improving the fit of a pool of reusable  
environments to requirements of programs in a computer  
system. The method comprises steps of: providing a first  
environment for a first program; responsive to initiation  
of a second program, making a determination whether  
creation of a new environment is a best response;  
responsive to a determination that creation of a new  
environment is a best response, creating a new environment  
for the second program; responsive to a determination that  
creating a new environment is not a best response, testing  
the pool for a best fit environment; and adding elements to  
the best fit environment to match requirements of the  
second program, unless the best fit environment already  
matches the requirements of the second program.

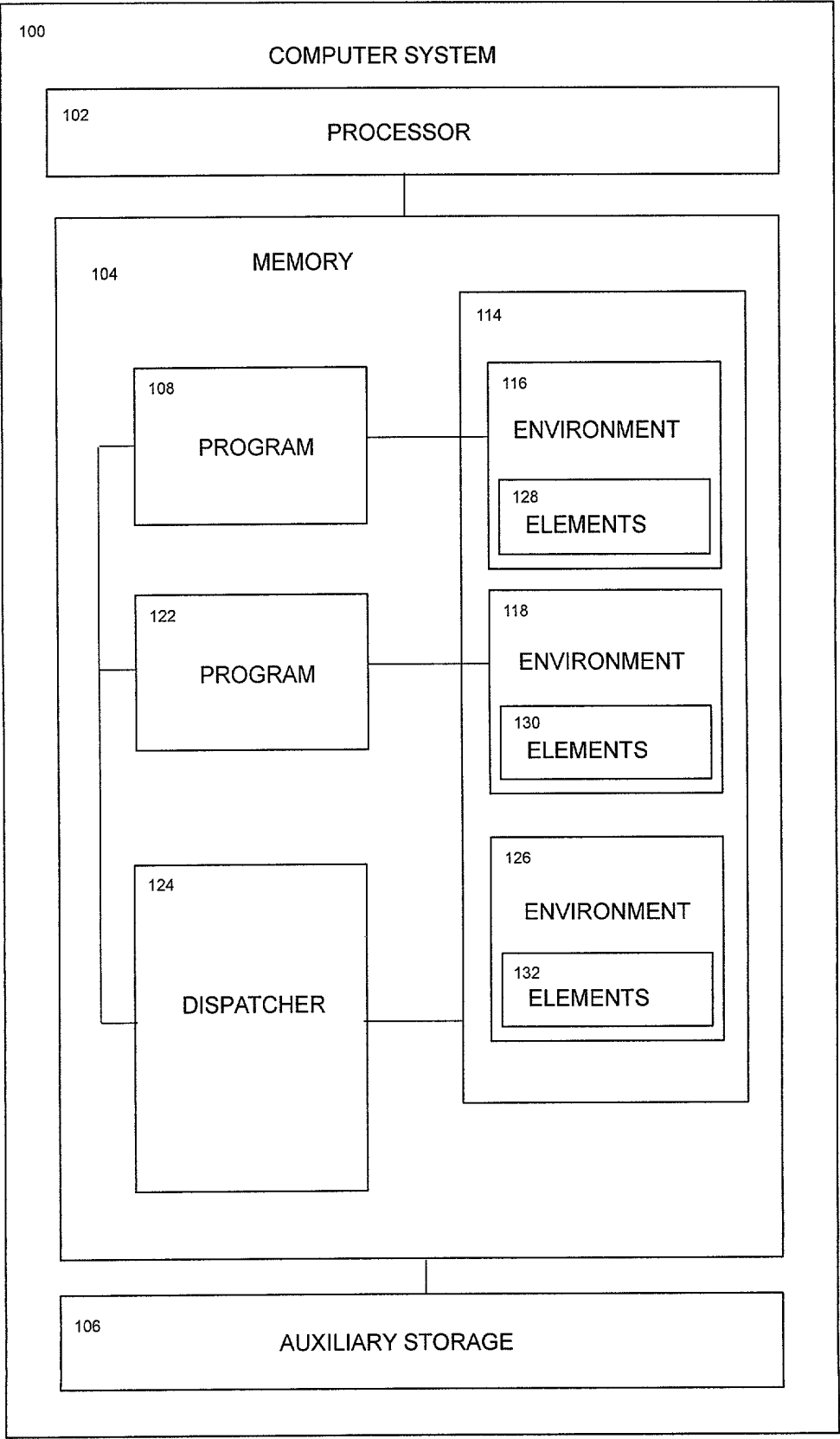


Figure 1

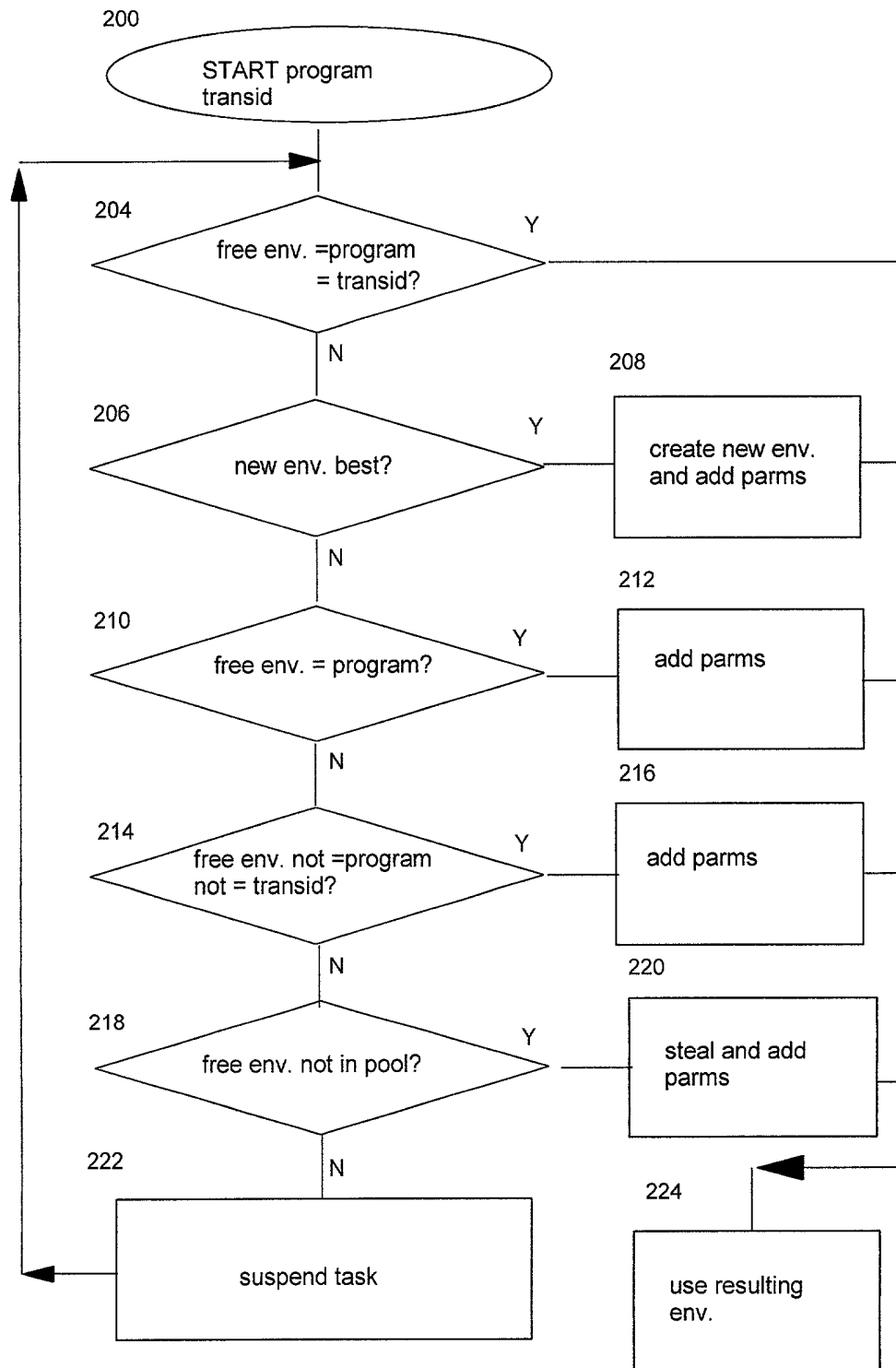


Figure 2

## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

### METHOD, PROGRAM PRODUCT AND COMPUTER SYSTEM FOR PROGRESSIVE IMPROVEMENT OF AN ENVIRONMENT POOL

the specification of which (check one)

  X   is attached hereto.

       was filed on  
as Application Serial No.  
and was amended on

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, CFR 1.56.

I hereby claim foreign priority benefits under Title 35, USC 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Number	Country	Filing Date Day/Month/Year	<u>      </u> Yes <u>  X  </u> No Priority Claimed
--------	---------	-------------------------------	---

I hereby claim the benefit under Title 35, USC 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, USC 112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, CFR 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Applications:

Serial No.	Filing Date	Status
------------	-------------	--------

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

001620 293350



**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Edward H. Duffield, Reg. No. 25,970; A. Bruce Clay, Reg. No. 32,121; Gregory M. Doudnikoff, Reg. No. 32,847; Jeanine S. Ray-Yarletts, Reg. No. 39,808; Jerry W. Herndon, Reg. No. 27,901, Kevin J. Fournier, Reg. No. 34,333

Send all correspondence to:

E H Duffield  
IBM Corp, IP Law Dept T81/B062,  
3039 Cornwallis Road,  
PO Box 12195,  
Research Triangle Park, NC 27709-2195.

---

FULL NAME OF SOLE OR FIRST INVENTOR: **Michael Casson BAILEY**

INVENTOR'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

RESIDENCE: 4 Grove Road, Shawford, Winchester, Hampshire SO21 2DD, United Kingdom

CITIZENSHIP: British

POST OFFICE ADDRESS: As above

FULL NAME OF SECOND INVENTOR: **Timothy William BANKS**

INVENTOR'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

RESIDENCE: 1 School Road, Twyford, Winchester, SO21 1QQ, United Kingdom

CITIZENSHIP: British

POST OFFICE ADDRESS: As above

FULL NAME OF THIRD INVENTOR: **Geoffrey REDDING**

INVENTOR'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

RESIDENCE: Home Mead, High Street, Twyford, Winchester, Hampshire SO21 1NW, United Kingdom

CITIZENSHIP: British

POST OFFICE ADDRESS: As above

007620 2983560

FULL NAME OF FOURTH INVENTOR: **Dennis J Zimmer**

INVENTOR'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

RESIDENCE: 17 Queens Road, Chandlers Ford, Eastleigh, Hampshire  
SO53 5AH, United Kingdom

CITIZENSHIP: British

POST OFFICE ADDRESS: As above